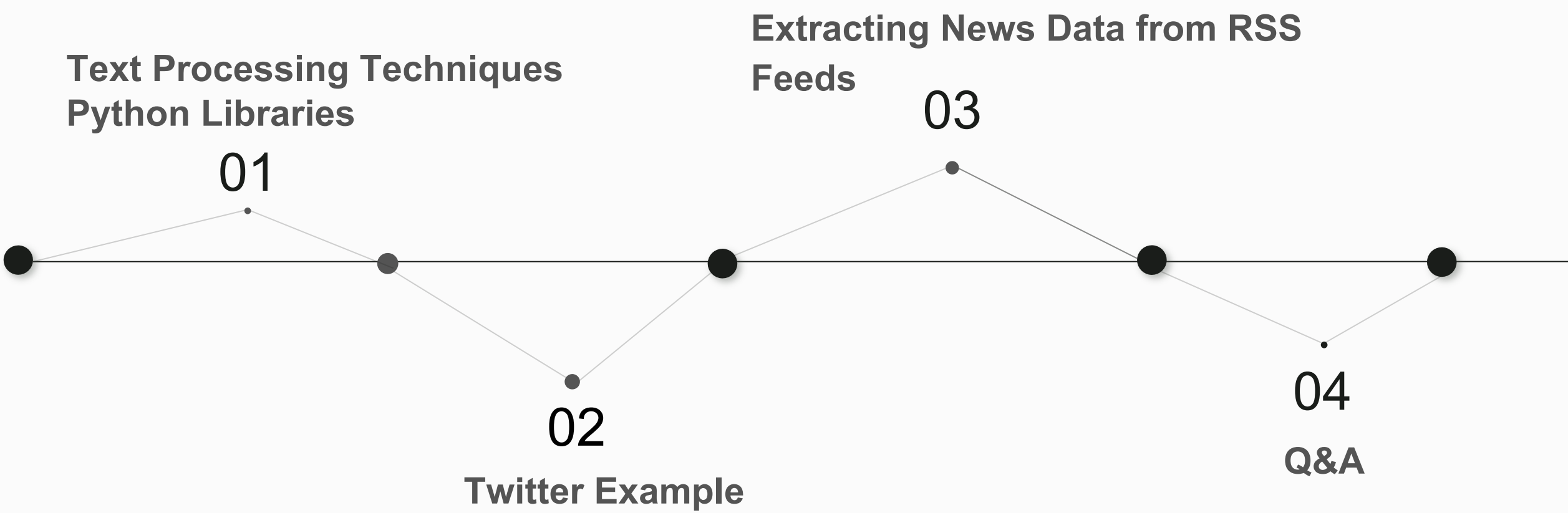
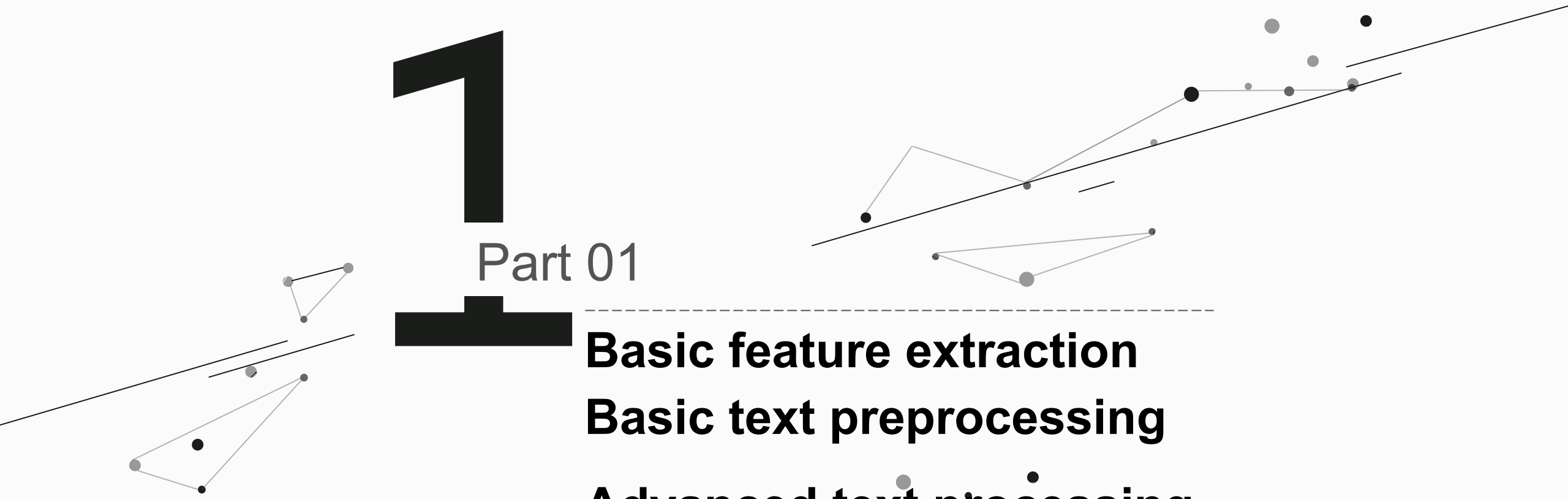
MIE1624

Text Processing

Reporter ：Group 16

CONTENT


Text Processing Techniques
Python Libraries
01
02
Twitter Example
Extracting News Data from RSS Feeds
03
04
Q&A



1

Part 01

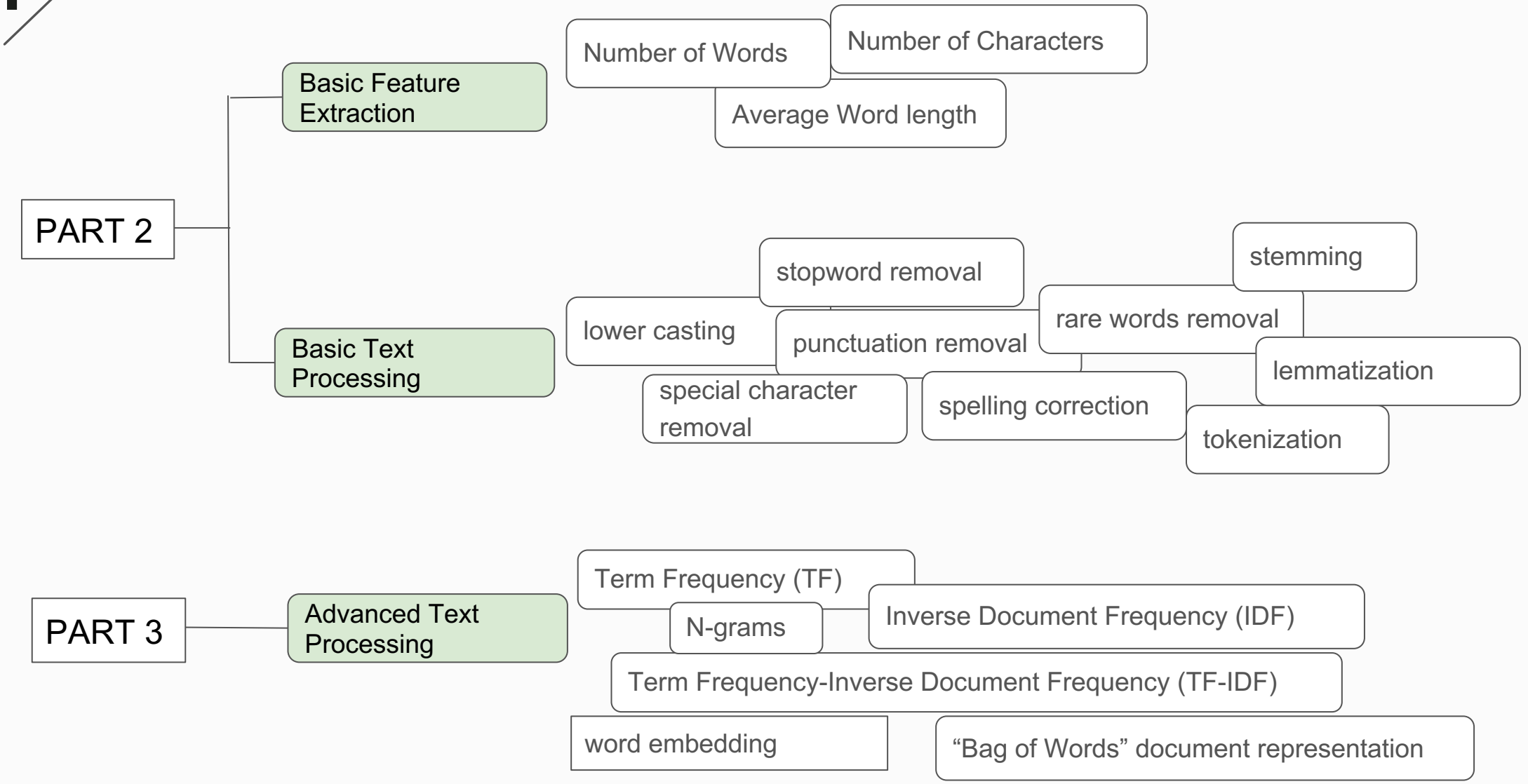
Basic feature extraction

Basic text preprocessing

Advanced text processing

Used Libraries

01


PART 2
Basic Feature Extraction
Number of Words
Number of Characters
Average Word length
Basic Text Processing
stopword removal
stemming
lower casing
rare words removal
punctuation removal
lemmatization
special character removal
spelling correction
tokenization
PART 3
Advanced Text Processing
Term Frequency (TF)
N-grams
Inverse Document Frequency (IDF)
Term Frequency-Inverse Document Frequency (TF-IDF)
word embedding
“Bag of Words” document representation

01 Library

- ***nltk***
  Natural Language Toolkit, provides easy-to-use interfaces to over 50 corpora and lexical resources.

- ***scikit learn***
  Simple and efficient tools for predictive data analysis, built on NumPy, SciPy, and matplotlib.

- ***re***
  Regular expression operations,provides regular expression matching operations similar to those found in Perl.

- ***beautifulsoup***
  For pulling data out of HTML and XML files.

- ***Request***
  The requests module allows you to send HTTP requests using Python

Part 02

2

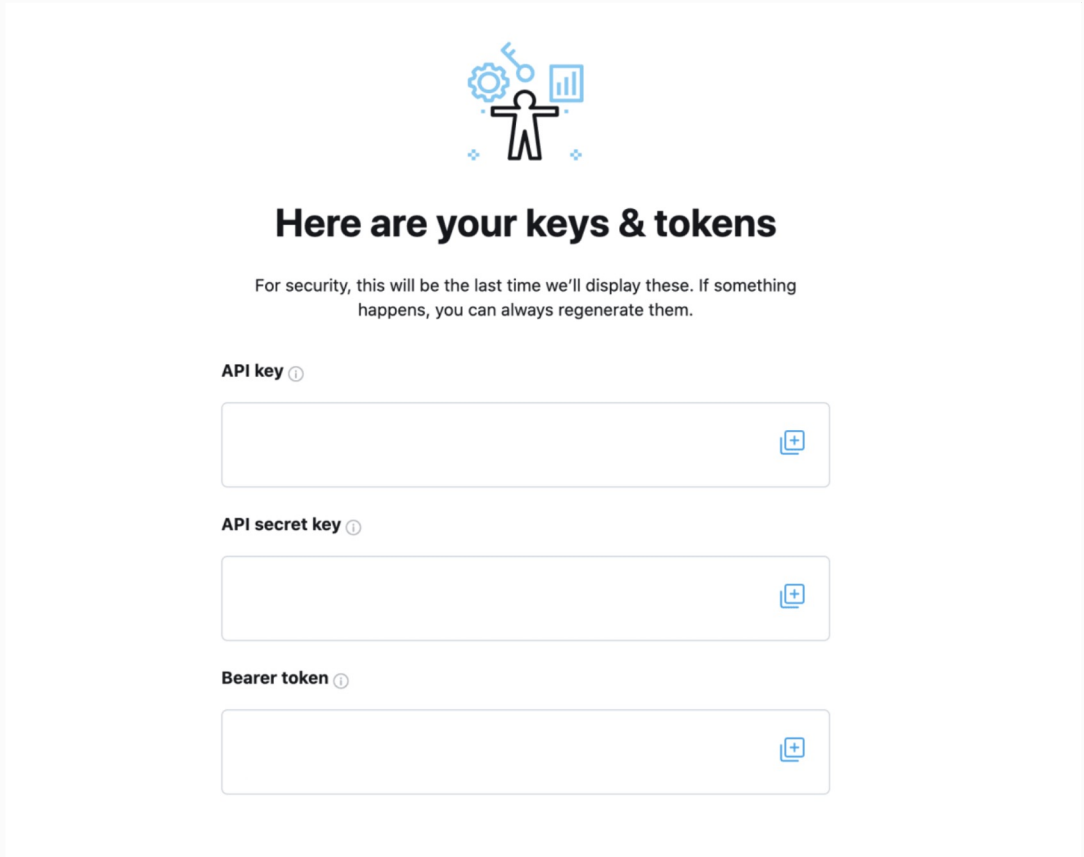
Twitter Example

02

1. Steps to get Twitter data in Python
2. Python modules and data structure
3. Limitations of the Twitter API
4. Alternative ways to get Twitter data
5. Jupyter Notebook example presentation

02

1 Steps to get Twitter data in Python



Twitter account

You need an approved developer account of Twitter, and then you will have the API keys and tokens from a developer App that is located within a Project.
(keep them as secret !!)

https://developer.twitter.com/en/docs/tutorials/step-by-step-guide-to-making-your-first-request-to-the-twitter-api-v2

02 1 Steps to get Twitter data in Python

To access Twitter, you will need to authenticate your account using your API keys and tokens.

We added our credentials to a txt file in advance. We will read the keys from the txt file.

```
# read twitter authorication keys from txt file
keys = []
with open('Twitter_Keys.txt') as f:
    for line in f:
        keys.append(line.strip())
```

```
consumer_key=keys[0] # consumer_key = 'YourConsumerKey'
consumer_secret=keys[1] # consumer_secret = 'YourConsumerSecret'
access_token=keys[2] # access_token = 'YourAccessToken'
access_token_secret=keys[3] # access_token_secret = 'YourAccessTokenSecret'
bearer_token = keys[4] # bearer_token = 'YourBearerToken'
```

```
# Authenicate your Account
client = tweepy.Client(consumer_key=consumer_key,
                       consumer_secret=consumer_secret,
                       access_token=access_token,
                       access_token_secret = access_token_secret,
                       bearer_token=bearer_token,
                       wait_on_rate_limit=True)
```

Twitter account

Firstly, you need an approved developer account on Twitter, and then authenticate using the keys and tokens from a developer App that is located within a Project

Authenticate your account in python

To access Twitter, you will need to authenticate your account using your API keys and tokens. We do this by adding our credentials to a python file. And the tweepy library allows you to interact with the Twitter API directly from Python and pull information in the form of JSON files. Use tweepy to authenticate your account in python.

02 1 Steps to get Twitter data in Python

```
uoft_search = client.search_recent_tweets(query="#uoft", max_results=10)
print(uoft_search)
type(uoft_search)
```

Twitter account

Firstly, you need an approved developer account on Twitter, and then authenticate using the keys and tokens from a developer App that is located within a Project

Authenticate your account in python

To access Twitter, you will need to authenticate your account using your API keys and tokens. We do this by adding our credentials to a python file. And the tweepy library allows you to interact with the Twitter API directly from Python and pull information in the form of JSON files. Use tweepy to authenticate your account in python.

Make a Call

We can make a call. For instance, we can use function search_all_tweets to search tweets made by accounts. The query, the parameter of the function, can be anything you are interested in ie. '#uoft'. This function will give JSON files containing metadata information about the tweets that were sent out.

02 1 Steps to get Twitter data in Python

```
uoft_search = client.search_recent_tweets(
    query="#uoft -is:retweet lang:en", # Extract non-retweeted English tweets
    max_results=100,
    expansions=["author_id"],
    tweet_fields= ["created_at,public_metrics"])
```

Twitter account

Firstly, you need an approved developer account on Twitter, and then authenticate using the keys and tokens from a developer App that is located within a Project

Authenticate your account in python

To access Twitter, you will need to authenticate your account using your API keys and tokens. We do this by adding our credentials to a python file. And the tweepy library allows you to interact with the Twitter API directly from Python and pull information in the form of JSON files. Use tweepy to authenticate your account in python.

Make a Call

We can make a call. For instance, we can use the function search_all_tweets to search tweets made by accounts. The query, the parameter of the function, can be anything you are interested in ie. '#uoft'. This function will give JSON files containing metadata information about the tweets that were sent out. So in order to view the data, we can use for loop to go through the data.

More details

If you want more details other than tweets or tweets id, then we can use Expansion or tweets fields etc, to expand the information included in the metadata. ie. we can retrieve the information about the author's id. You can include as much information as you would like to include including likes and reply_to.

02 / 1 Steps to get Twitter data in Python

```
# create our data set
data = []

#set the columns
columns = ['ID', 'Tweet', "Date Posted",'Author ID', 'Liked', 'Reply', 'Retweet']

# create a dictionary that will use the author_id field to look up more information
# about the users
uoft_users = {user['id']:
    user for user in uoft_search.includes['users']}

#add the data from our retieval to the data set
for tweet in uoft_search.data:
    if uoft_users[tweet.author_id]:
        user = uoft_users[tweet.author_id]
        data.append([tweet.id,
                     tweet.text,
                     tweet.created_at,
                     user.username,
                     tweet.public_metrics['like_count'],
                     tweet.public_metrics['reply_count'],
                     tweet.public_metrics['retweet_count']])

#create the dataframe
uoft_df = pd.DataFrame(data, columns=columns )
```

```
# export the data as csv
uoft_df.to_csv("uoft_tweets_current.csv")
```

```
# read we pre saved uoft_tweets_Nov13.csv to run the following steps
uoft_df = pd.read_csv('uoft_tweets_Nov13.csv')
```

Twitter account

Firstly, you need an approved developer account of Twitter, and then authenticate using the keys and tokens from a developer App that is located within a Project

Authenticate your account in python

To access Twitter, you will need to authenticate your account using your API keys and tokens. We do this by adding our credentials to a python file. And the tweepy library allows you to interact with the Twitter API directly from Python and pull information in the form of JSON files. Use tweepy to authenticate your account in python.

Make a Call

We can make a call. For instance, we can use function search_all_tweets to search tweets made by accounts. The query, the parameter of the function, can be anything you are interested in ie. ‘university of toronto’. This function will give JSON files containing metadata information about the tweets that were sent out. So in order to view the data, we can use for loop to go through the data.

Expansion

Then we can use Expansion, to expand the information included in the metadata. ie we can retrieve the information about author id. You can include as much as information that you would like to include including likes, reply_to.

To CSV file

Finally, we can use the information to generate the data frame and then do some of the text cleanings. And then you may convert the data frame to a csv file.

Python modules and data structure

Python

- **tweepy** Twitter for Python
- **twarc** a command line tool and Python library for collecting JSON data via the Twitter API, with a command (twarc2) for working with the v2 API
- **python-twitter** a simple Python wrapper for Twitter API v2
- **TwitterAPI** minimal Python wrapper for Twitter's APIs
- **twitterati** Wrapper for Twitter Developer API V2
- **twitter-stream.py** a Python API client for Twitter API v2
- **twitivity** Account Activity API client library for Python
- **PyTweet** a synchronous Python wrapper for the Twitter API
- **tweetkit** a Python Client for the Twitter API for Academic Research
- **tweetple** a wrapper to stream information from the Full-Archive Search Endpoint, for Academic Research
- **2wttr** get Tweets from the v2 Twitter API, for Academic Research

(https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries/v2)

Built by members from Twitter developer community.

Tweepy:

Enable us to interact with Twitter API.

Performs complex queries in addition to scraping tweets.

Enables you to take advantage of all of the Twitter API's capabilities.

Python modules and data structure

Search Tweets

Client.search_all_tweets(*query, *, end_time=None, expansions=None, max_results=None, media_fields=None, next_token=None, place_fields=None, poll_fields=None, since_id=None, sort_order=None, start_time=None, tweet_fields=None, until_id=None, user_fields=None*)

Client.search_recent_tweets(*query, *, end_time=None, expansions=None, max_results=None, media_fields=None, next_token=None, place_fields=None, poll_fields=None, since_id=None, sort_order=None, start_time=None, tweet_fields=None, until_id=None, user_fields=None, user_auth=False*)

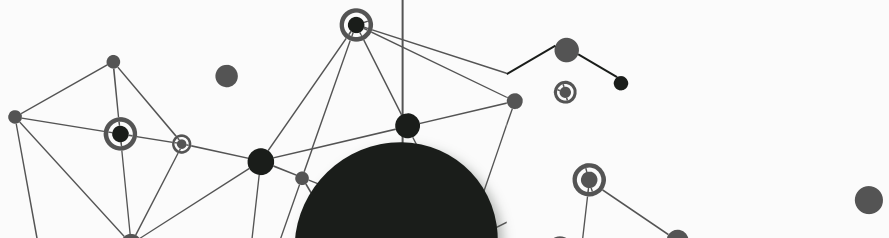
The recent search endpoint returns Tweets from the last seven days that match a search query.

- Either of the two returns us a **dictionary**.
- **No size limitation** to a dictionary in Python, except the capacity of your available memory.

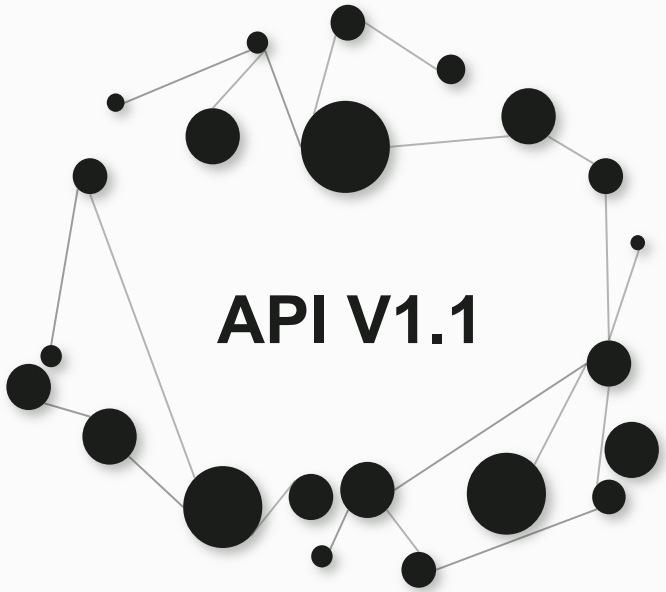
Want to present the information as a table or a csv.file in the further step?

In the example above, we chose to store the information of each tweet as a list, which was appended to the empty list created before (called ‘data’ in our example)

With the help of pandas ⇒ table & csv. file.



Limitations of the Twitter API



Query abilities on searching tweets using standard v1.1: limited for the past 7 days.

Recent search / Full-archive search

Limitations of the Twitter API

- Rate Limits (https://developer.twitter.com/en/docs/twitter-api/rate-limits)
- Limitation of Queries

Resource	Endpoint	Requests per 15-minute window unless otherwise stated	
		Per App	Per user
Tweets	Tweet lookup	300	900
	Manage Tweets - Post a Tweet - Delete a Tweet		 200 50
	Timelines - User Tweet timeline - User mention timeline - Reverse chronological home timeline	 1500 450	 900 180 180
	Search Tweets - Recent search - Full-archive search	 450 300 Full-archive also has a 1 request / 1 second limit	180

02 3

Limitations of the Twitter API

- Rate Limits (https://developer.twitter.com/en/docs/twitter-api/rate-limits)
- Limitation of Queries

Essential or Elevated access: query can be 512 characters long.
Academic Research access: query can be 1024 characters long.

Alternative way to get Twitter data

Snscrape

- Do NOT need API
- Scrape basic information, such as users, user profiles, hashtags, searches, tweets, list posts, and trends.
- No limit to the number of tweets you can fetch
- Can be used on other prominent social media networks like Facebook, Instagram,etc.

02 5 Jupyter Notebook example presentation

1. Use Twitter API for python to **downloads** tweets. Save those as a **csv**. file

2. Perform **basic feature extraction** & **basic text preprocessing** on tweets from the csv. file.

3. Perform the alternative way by **Snscrape**

02

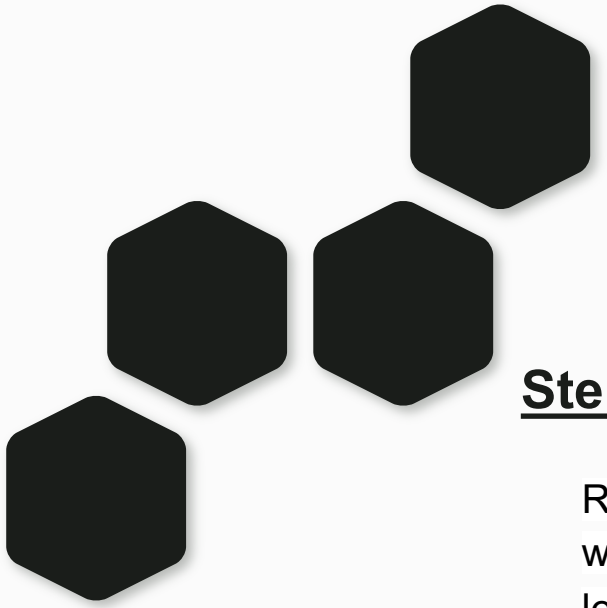
If you are not familiar with them …

Tokenization

Breaking up the paragraph into smaller units such as sentences or words. Each unit is then considered as an individual token. Try to understand the meaning of the text by analyzing the smaller units or tokens that constitute the paragraph.

Stopwords removal

Collection of words that occur frequently but do not add much meaning to the sentences. e.g. Some English stop words: “the”, “he”, “him”, “his”, “her”, “herself” etc.



Lemmatization

Lemmatization considers the context and converts the word to its meaningful base form, which is called Lemma. e.g. The words “studying”, “studies”, “study” are all reduced to “study”.

Stemming

Reduction of a word into its root or stem word. The word affixes are removed leaving behind only the root form or lemma. e.g. The words “studying”, “studies”, “study” are all reduced to “studi”.

Part 3: RSS feeds (Really Simple Syndication)

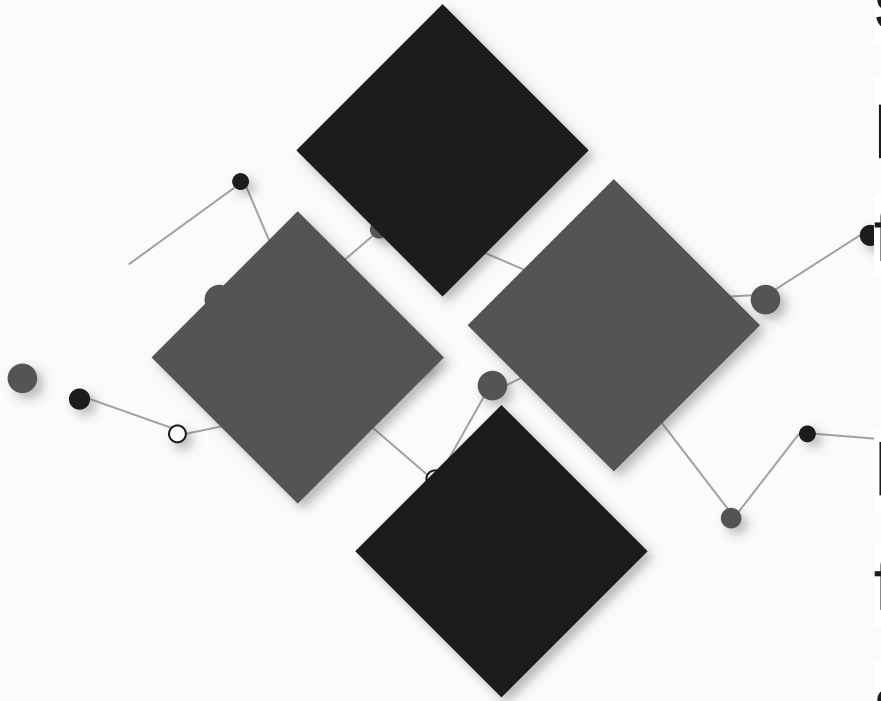
XML-based format:

Sharing and distributing Web content, such as news headlines, authors, publication date.

View data feeds from many news sources.

03 What is Feedparser?

Feedparser:
a Python library that parses feeds in all known formats

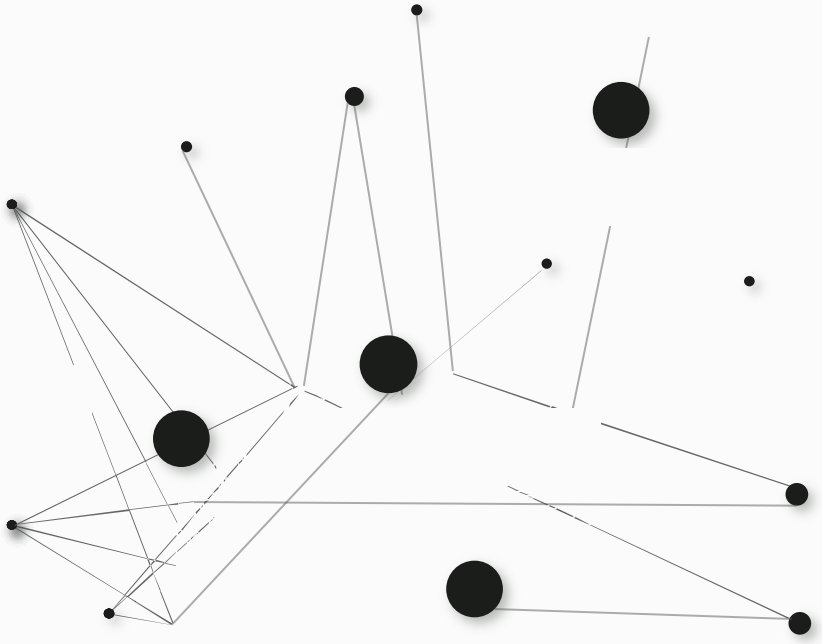


Feedparser:
can be used to extract information about a specific webpage or a publication with its RSS feed(not only RSS).

By providing the RSS feed link, we can get structured information in the form of python lists and dictionaries.

03

What is bag of words



Transforms the text into fixed-length vectors

Count the number of times the word is present in a document

compare different documents and evaluate their similarities for applications

03

TF and TF-IDF

Term Frequency (tf):
Frequency of the word

counting the number of times a word appears

TF

TF-IDF

Term frequency-inverse document frequency (TF-IDF)

a numerical weightage of words

used for information retrieval and text mining.

Code

```
rawrss = [
    #this RSS feed is from the BBC NEWS
    'http://newsrss.bbc.co.uk/rss/newsonline_uk_edition/front_page/rss.xml'
    ]


posts = []

##first to loop through all “links” in the RSS feed and parsing feed using python library feedparser at each iteration
for url in rawrss:
    feed = feedparser.parse(url)

    for post in feed.entries:

        #Request the article url to get the web page content and create a BeautifulSoup object with the HTML from that page
        article = requests.get(post.link)
        articles = BeautifulSoup(article.content, 'html.parser')

        #extract all paragraph elements inside the page body, and for each paragraph , extract its element text and append it to a list.
        articles_body = articles.findAll('body')
        p_blocks = articles_body[0].findAll('p')

        body=[]
        # Loop trough paragraph to extract its element text
        for i in range(0,len(p_blocks)):

          body.append(p_blocks[i].text)
        #unpack list
        body=''.join(body)

        #in each post, save its link, title, description and text.
        posts.append((post.title, post.link, post.description, body))

#create a Pandas dataframe from RSS parsing results above with title, link, description and text of all news articles
df = pd.DataFrame(posts, columns=['title', 'link','description','Text'])

df.to_csv('newsdataframe.csv')
```

Output

```
df.head()
```

	title	link	description	Text	# of words	# of characters	# of stopwords	Average word length
0	UK strikes revised deal with France on Channel...	https://www.bbc.co.uk/news/uk-politics-6361565...	The UK will pay France £8m more a year to incr...	This video can not be playedWatch: Suella Brav...	1036	6299	72	5.081081
1	COP27: War causing huge release of climate war...	https://www.bbc.co.uk/news/science-environment...	Ukraine tells UN climate summit it will use ev...	Russia's invasion of Ukraine has caused a larg...	527	3313	49	5.288425
2	Xi Biden meeting: US leader promises 'no new C...	https://www.bbc.co.uk/news/world-asia-63628454...	The leaders of the US and China strike a conci...	This video can not be playedWATCH: Biden says ...	975	5860	70	5.010256
3	Jonnie Irwin: Place in the Sun presenter revea...	https://www.bbc.co.uk/news/entertainment-arts-...	The Place in the Sun and Escape to the Country...	TV presenter Jonnie Irwin has revealed he has ...	779	4374	74	4.616175
4	Princess Anne and Prince Edward to become stan...	https://www.bbc.co.uk/news/uk-63626113?at_medi...	The King requests extra stand-ins, as Prince A...	King Charles has begun the process of increasi...	641	3955	53	5.171607

TF (count vectorization)

```
#Bag-of-words using count vectorization (TF)
from sklearn.feature_extraction.text import CountVectorizer
#token_pattern=r'\b[a-zA-Z]{3,}\b' exclude anything that has numbers in it.
vectorizer = CountVectorizer(analyzer='word', token_pattern=r'\b[a-zA-Z]{3,}\b',stop_words='english')
X = vectorizer.fit_transform(df['Text'])
#in matrix form
tf=pd.DataFrame(X.toarray(),
                columns=vectorizer.get_feature_names())
tf.head()
```

```
import nltk
from nltk.corpus import stopwords
print(stopwords.words('english'))
```

Stopwords

{'ourselves', 'hers', 'between', 'yourself', 'but', 'again', 'there', 'about', 'once', 'during', 'out', 'very', 'having', 'with', 'they', 'own', 'an', 'be', 'some', 'for', 'do', 'its', 'yours', 'such', 'into', 'of', 'most', 'itself', 'other', 'off', 'is', 's', 'am', 'or', 'who', 'as', 'from', 'him', 'each', 'the', 'themselves', 'until', 'below', 'are', 'we', 'these', 'your', 'his', 'through', 'don', 'nor', 'me', 'were', 'her', 'more', 'himself', 'this', 'down', 'should', 'our', 'their', 'while', 'above', 'both', 'up', 'to', 'ours', 'had', 'she', 'all', 'no', 'when', 'at', 'any', 'before', 'them', 'same', 'and', 'been', 'have', 'in', 'will', 'on', 'does', 'yourselves', 'then', 'that', 'because', 'what', 'over', 'why', 'so', 'can', 'did', 'not', 'now', 'under', 'he', 'you', 'herself', 'has', 'just', 'where', 'too', 'only', 'myself', 'which', 'those', 'i', 'after', 'few', 'whom', 't', 'being', 'if', 'theirs', 'my', 'against', 'a', 'by', 'doing', 'it', 'how', 'further', 'was', 'here', 'than'}

output

	aberdeen	aberdeenshire	abigail	ability	able	abolished	abroad	absence	absolutely	absorbing	...	zealandavailable	zelensky	zero	zest	zhou	zlotys	zone	zon
0	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
1	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	1	0	...	0	0	1	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	
4	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	

5 rows × 6247 columns

TF_IDF (term frequency–inverse document frequency)

```
#Bag-of-words using TF_IDF
from sklearn.feature_extraction.text import TfidfVectorizer
#token_pattern=r'\b[a-zA-Z]{3,}\b' exclude anything that has numbers in it.
vectorizer = TfidfVectorizer(analyzer='word', token_pattern=r'\b[a-zA-Z]{3,}\b',stop_words='english')
X = vectorizer.fit_transform(df['Text'])
#in matrix form
tf_idf=pd.DataFrame(X.toarray(),
             columns=vectorizer.get_feature_names())
tf_idf.head()
```

Output

	aberdeen	aberdeenshire	abigail	ability	able	abolished	abroad	absence	absolutely	absorbing	...	zealandavailable	zelensky	zero	zest	zhou	zloty
0	0.0	0.0	0.0	0.0	0.024015	0.0	0.000000	0.0	0.000000	0.0	...	0.0	0.0	0.000000	0.000000	0.0	0
1	0.0	0.0	0.0	0.0	0.039157	0.0	0.000000	0.0	0.000000	0.0	...	0.0	0.0	0.000000	0.000000	0.0	0
2	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.029994	0.0	...	0.0	0.0	0.025615	0.000000	0.0	0
3	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.000000	0.0	...	0.0	0.0	0.000000	0.052077	0.0	0
4	0.0	0.0	0.0	0.0	0.000000	0.0	0.034136	0.0	0.000000	0.0	...	0.0	0.0	0.000000	0.000000	0.0	0

5 rows × 6247 columns

Reference

- https://developer.twitter.com/en/docs
- https://docs.tweepy.org/en/stable/index.html
- https://github.com/mattwvu/workshop_python_twitter_fall2022/blob/main/scrape_twitter_tweepy.ipynb
- https://www.freecodecamp.org/news/python-web-scraping-tutorial/
- https://www.nltk.org/index.html
- https://medium.com/analytics-vidhya/web-scraping-news-data-rss-feeds-python-and-google-cloud-platform-7a0df2bafe44
- https://www.analyticsvidhya.com/blog/2021/07/bag-of-words-vs-tfidf-vectorization-a-hands-on-tutorial/
- https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089
- https://www.kaggle.com/code/gauravahujaravenclaw/step-1-to-learn-nlp-text-pre-processing

Part 04

4 Q & A

Thank You